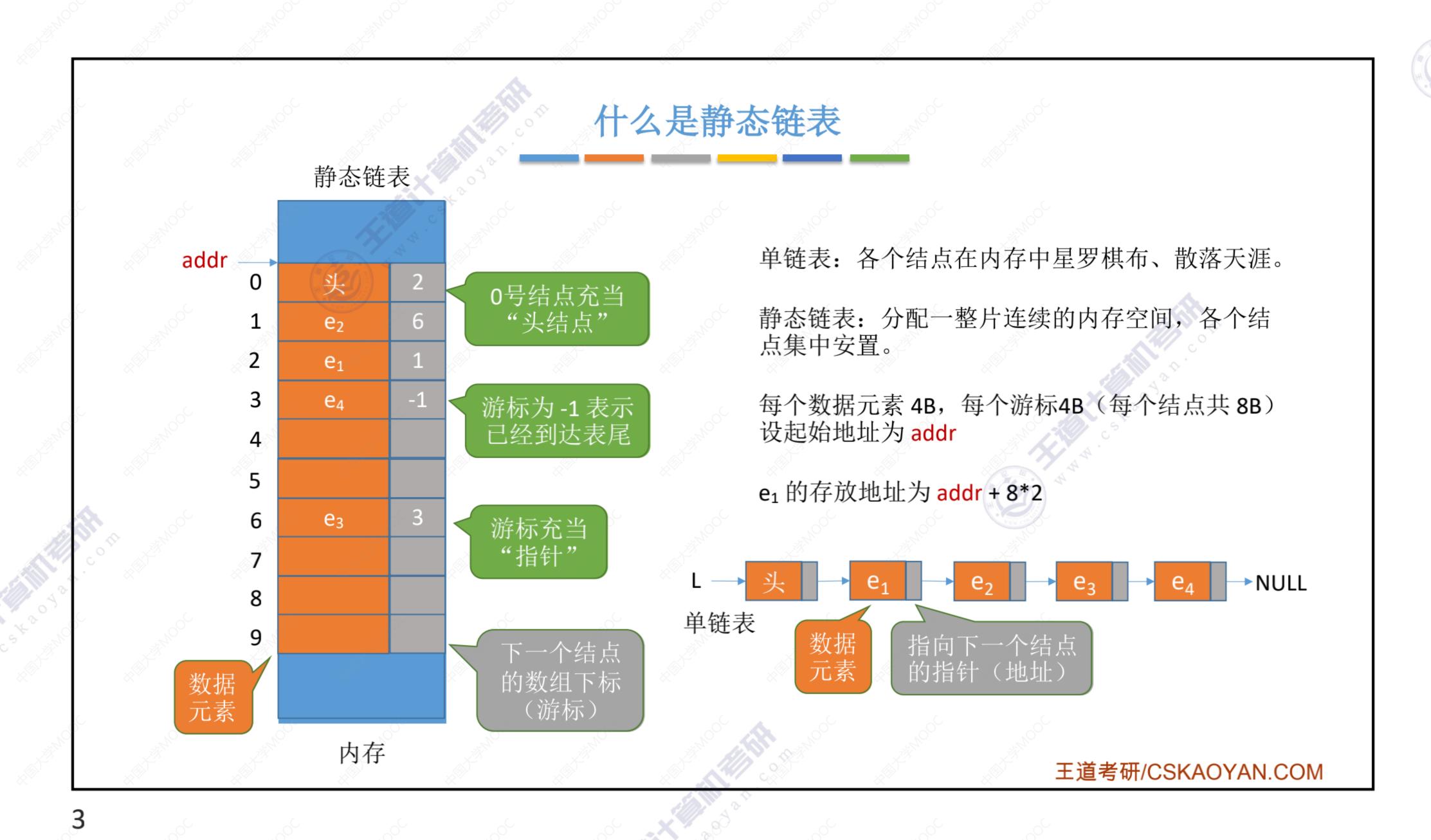
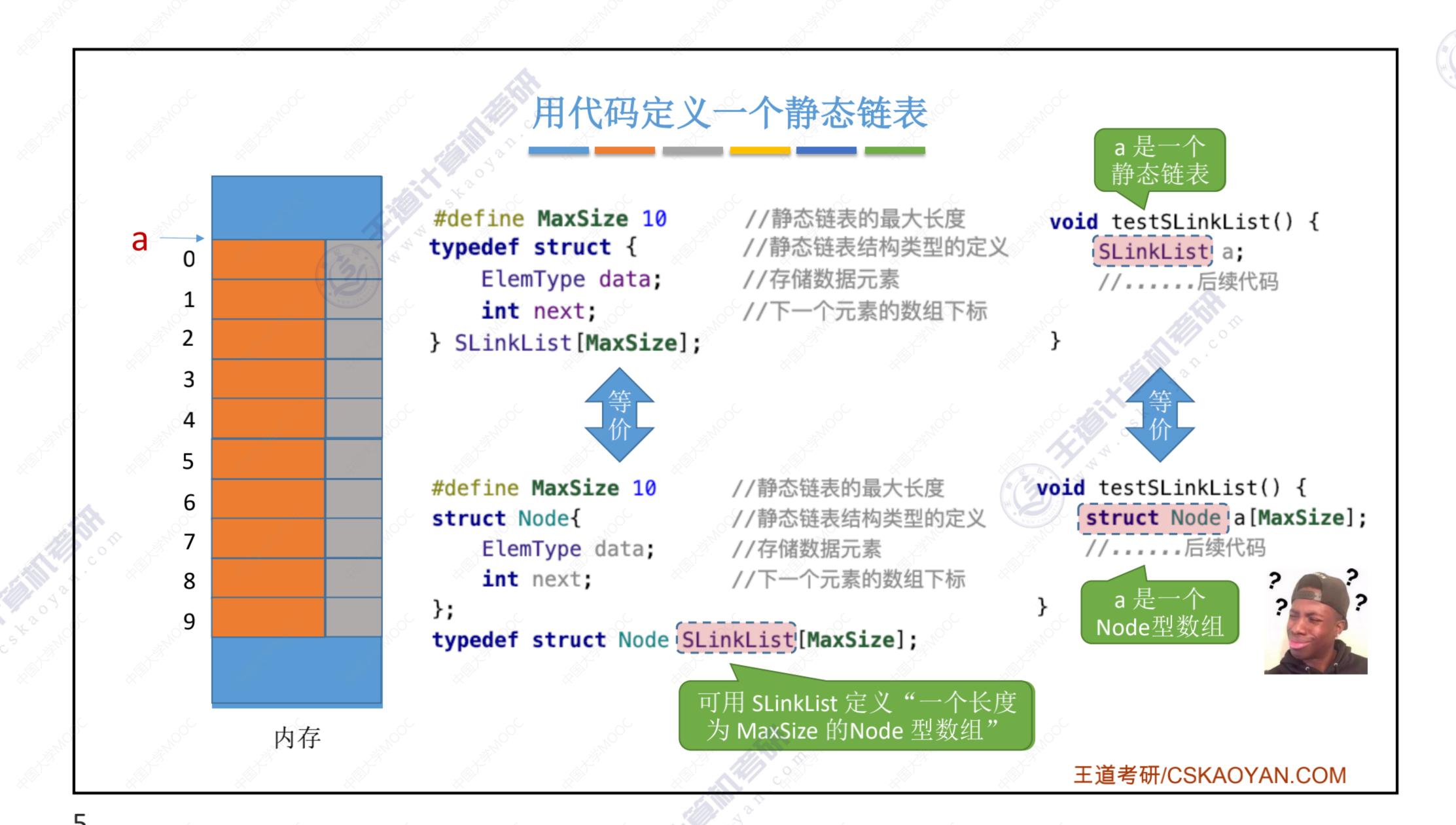


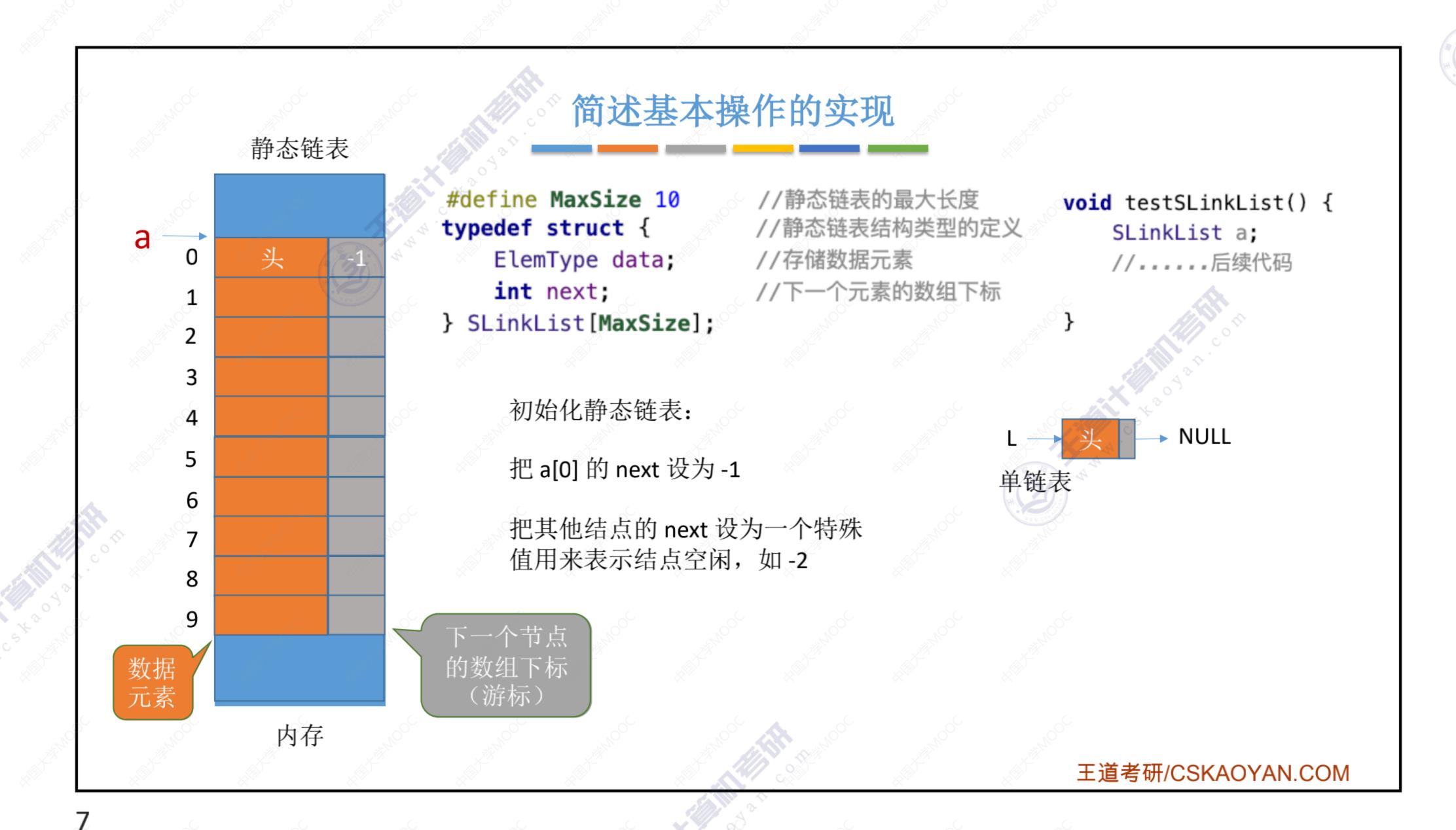
2

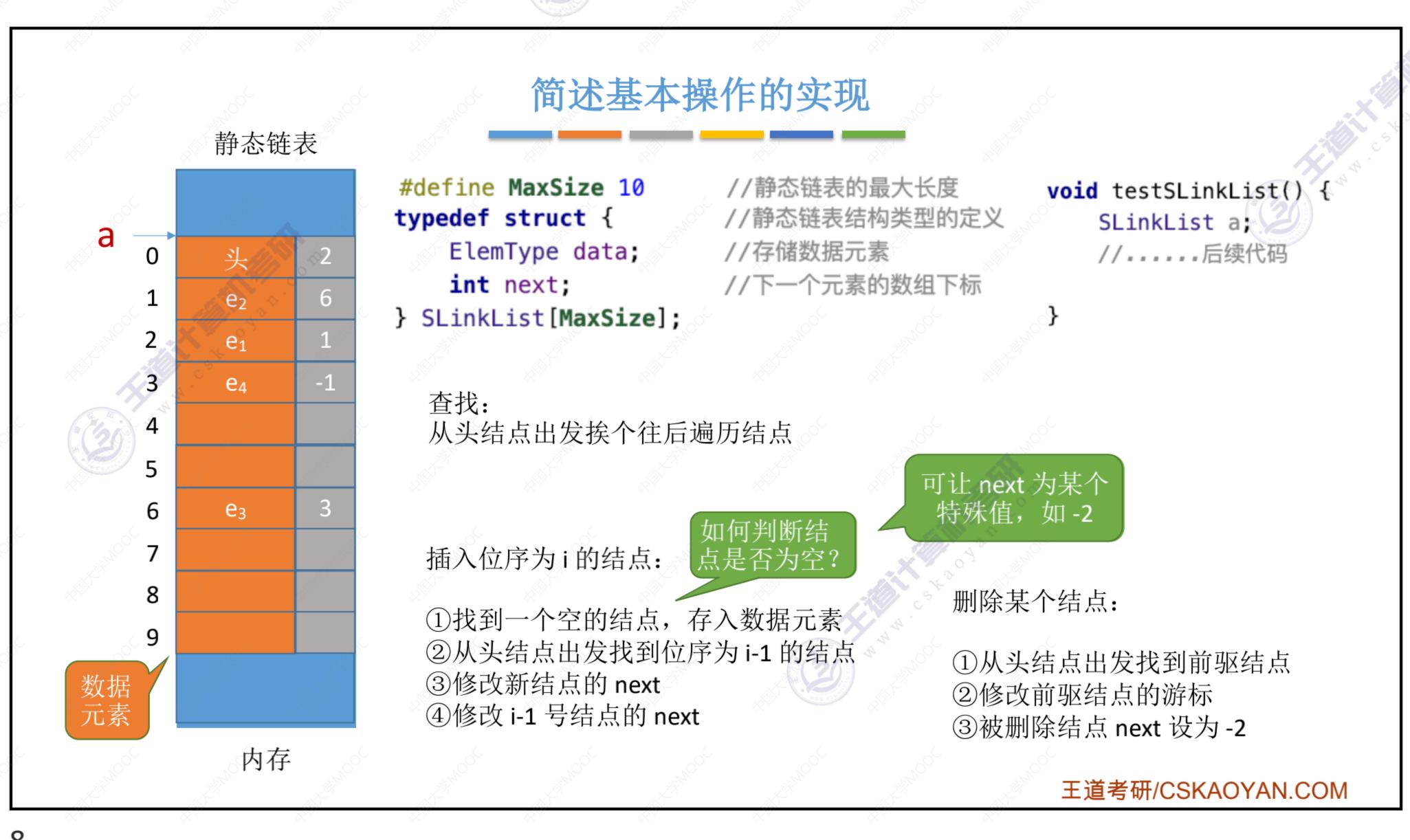


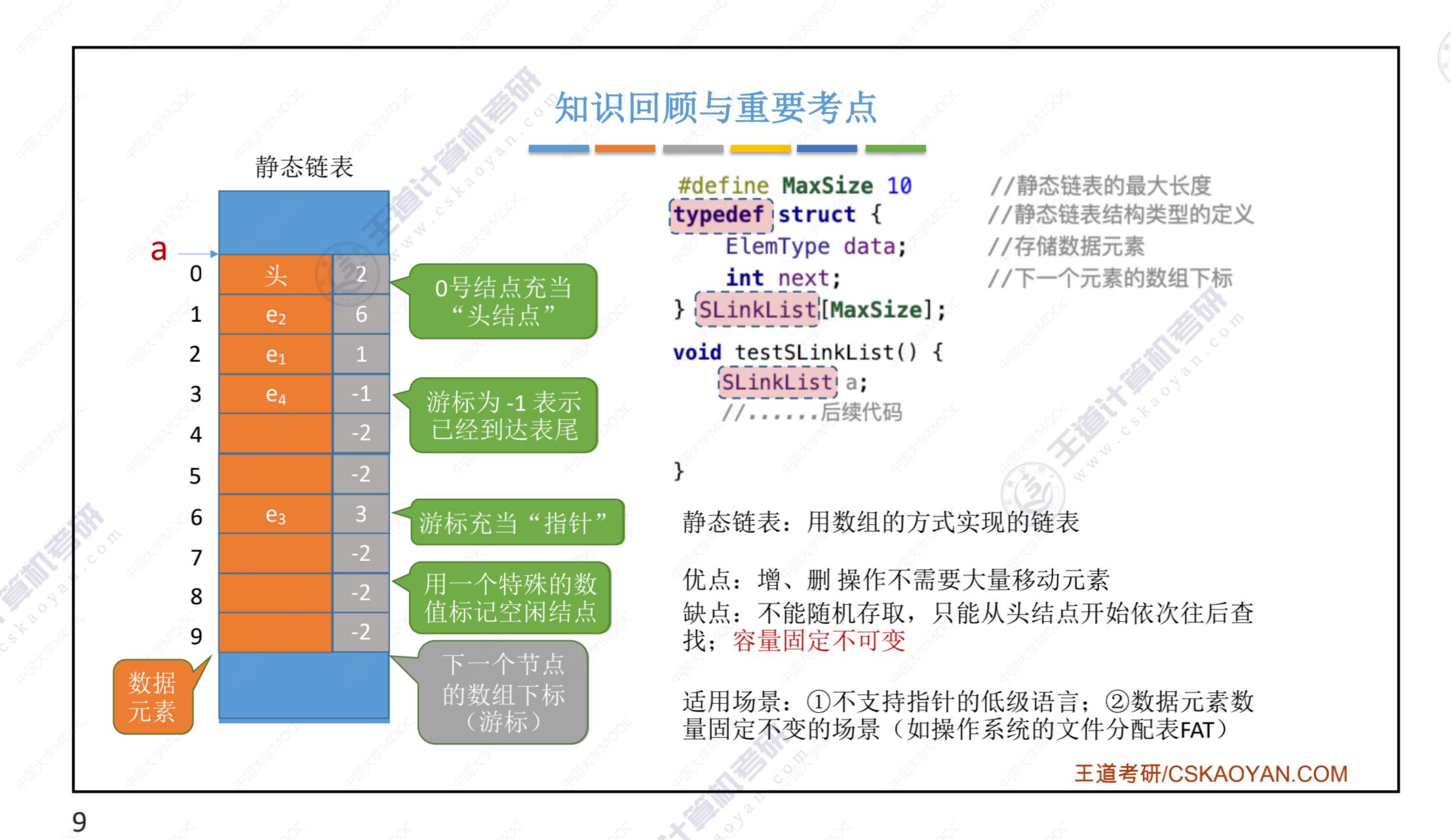
用代码定义一个静态链表 #define MaxSize 10 //静态链表的最大长度 a struct Node{ //静态链表结构类型的定义 //存储数据元素 ElemType data; int next; //下一个元素的数组下标 void testSLinkList() { 数组 a 作为 静态链表 struct Node a[MaxSize]; // 后续代码 next data (游标) (数据元素) 内存 王道考研/CSKAOYAN.COM



对猜想的验证 #define MaxSize 10 //静态链表的最大长度 struct Node{ //静态链表结构类型的定义 int data; //存储数据元素 int next; //下一个元素的数组下标 **}**; //静态链表结构类型的定义 typedef struct { 结论: int data; //存储数据元素 int next; //下一个元素的数组下标 SLinkList b —— 相当于定义了一个长度 } SLinkList[MaxSize]; 为 MaxSize 的 Node型数组 void testSLinkList() { struct Node x; printf("sizeX=%d\n", sizeof(x)); 运行结果 struct Node a[MaxSize]; sizeX=8 printf("sizeA=%d\n", sizeof(a)); sizeA=80 sizeB=80 SLinkList b; printf("sizeB=%d\n", sizeof(b)); Process finished with exit code 0 王道考研/CSKAOYAN.COM









10